Complexity of the cluster deletion problem on some subclasses of chordal graphs

Flavia Bonomo, Guillermo Duran, Mario Valencia-Pabon

▶ **To cite this version:**

Flavia Bonomo, Guillermo Duran, Mario Valencia-Pabon. Complexity of the cluster deletion problem on some subclasses of chordal graphs. Theoretical Computer Science, 2015, 600, pp.59-69. hal-01102512

HAL Id: hal-01102512
https://hal.science/hal-01102512

Submitted on 13 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Complexity of the cluster deletion problem on subclasses of chordal graphs and cographs*

Flavia Bonomo† Guillermo Duran‡ Mario Valencia-Pabon§

Abstract

We consider the following vertex-partition problem on graphs: given a graph with real nonnegative edge weights, partition the vertices into clusters (in this case cliques) to minimize the total weight of edges out of the clusters. This optimization problem is known to be an NP-complete problem even for unweighted graphs and has been studied extensively in the scope of fixed-parameter tractability (FPT), where it is commonly known as the CLUSTER DELETION problem. Many of the recently-developed FPT algorithms rely on being able to solve CLUSTER DELETION in polynomial-time on restricted graph structures. In this paper, the complexity of the CLUSTER DELETION problem is investigated for the family of chordal graphs. It is shown that this problem is NP-complete for edge-weighted split graphs, edge-weighted interval graphs and edge-unweighted chordal graphs. We also prove that the CLUSTER DELETION is an NP-complete problem for edge-weighted cographs. Some polynomial-time solvable cases are also identified, in particular CLUSTER DELETION for unweighted split graphs, unweighted proper interval graphs and weighted block graphs.

Keywords: Block graphs, Cliques, Edge-deletion, Cluster deletion, Interval graphs, Split graphs, Submodular functions, Chordal graphs, Cographs, NP-completeness.

1 Introduction

Clustering is an important task in process of data analysis which can be viewed as a data modeling technique that provides an attractive mechanism to automatically find the hidden structure of large data sets. The input to the problem is typically a set of elements and pairwise similarity values between elements. The goal is to partition the elements into subsets, which are called *clusters*, so that two meta-criteria are satisfied: *Homogeneity*–elements inside a cluster are highly similar to each other; and *separation*–elements from different clusters have low similarity to each other. In the graph theoretic approach to clustering, one builds from the raw data a *similarity graph* whose vertices corresponds to elements and there is an edge between two vertices if and only if

*Partially supported by MathAmSud Project 13MATH-07 (Argentina–Brazil–Chile–France), UBACyT Grant 20020100100980, CONICET PIP 112-200901-00178 and 112-201201-00450CO and ANPCyT PICT 2012-1324 (Argentina), FONDECyT Grant 1140787 and Millennium Science Institute "Complex Engineering Systems" (Chile).

†CONICET and Dep. de Computación, FCEN, Universidad de Buenos Aires, Argentina. e-mail: fbonomo@dc.uba.ar

‡CONICET and Dep. de Matemática and Instituto de Cálculo, FCEN, Universidad de Buenos Aires, Argentina, and Dep. de Ingeniería Industrial, FCFM, Universidad de Chile, Santiago, Chile. e-mail: gduran@dm.uba.ar

§Université Paris-13, Sorbonne Paris Cité LIPN, CNRS UMR7030, Villetaneuse, France. Currently in Délégation at the INRIA Nancy - Grand Est. e-mail: valencia@lipn.univ-paris13.fr

the similarity of their corresponding elements exceeds a predefined threshold [11, 12]. Cluster graphs have been used in a variety of applications whenever clustering of objects is studied or when consistent data is sought among noisy or error-prone data [1, 4]. Ideally, the resulting graph would be a *cluster graph*, that is, a graph in which every connected component is a clique (i.e. a complete subgraph). In practice, it is only close to being such, since similarity data is experimental and thus, error-prone.

The *cluster deletion* problem asks for the minimum number of edges that can be removed from an input graph to make the resulting graph a cluster graph. There exist several results for the cluster deletion problem (see for example [3, 17, 14] and references therein). The cluster deletion problem is known to be NP-complete [17] for general graphs. Moreover, Shamir et al. [17] showed that it remains NP-hard when enforcing that the input graph should be clustered into exactly $d \geq 3$ components. They also showed that when the input graph should be clustered into exactly 2 components, the problem is polynomial-time solvable. Komusiewicz et al. [14] proved that cluster deletion is hard for C_4-free graphs with maximum degree 4, and they gave a $O(n^{1.5} \log^2 n)$ time algorithm for solving cluster deletion on graphs with maximum degree 3. Based on results obtained by Demaine et al. [5] concerning a variant of a clustering problem, Dessmark et al. [6] provided a polynomial $O(\log n)$-approximation algorithm for the edge-weighted version of the cluster deletion problem. They also showed that for the edge-unweighted version of cluster deletion on general graphs, the greedy algorithm that finds iteratively maximum cliques gives a 2-approximation algorithm to the optimal cluster deletion. The complexity of such an algorithm relies on the complexity of iteratively finding maximum cliques, so it is a polynomial-time approximation algorithm for certain graph classes. Recently, Gao et al. [9] showed that the greedy algorithm that finds iteratively maximum cliques, gives an optimal solution for the class of graphs known as *cographs*. It implies that the cluster deletion problem is polynomial-time solvable on unweighted cographs. Moreover, Gao et al. [9] showed that the cluster deletion problem is NP-hard on (C_5, P_5)-free graphs, on $(2K_2, 3K_1)$-free graphs and on $(C_5, P_5, \text{bull}, \text{4-pan}, \text{fork}, \text{co-gem}, \text{co-4-pan})$-free graphs. For weighted graphs, the cluster deletion problem can be solved in polynomial time on the class of triangle-free graphs, since it is equivalent to maximum weighted matching [7]. The cluster deletion problem has been studied extensively in the scope of fixed-parameter tractability (FPT). Many of the recently-developed FPT algorithms rely on being able to solve cluster deletion in polynomial-time on restricted graph structures [3].

This paper is organized as follows. In Section 2, we prove that the weighted cluster deletion problem is NP-complete in a subclass of split graphs and even under certain restrictions for the weights, and that it is polynomial-time solvable for unweighted split graphs. In Section 3, we prove that the unweighted cluster deletion problem is NP-complete in a subclass of chordal graphs, namely P_5-free chordal graphs. In Section 4, we prove that the weighted cluster deletion problem is polynomial-time solvable on block graphs, when the weights are nonnegative. In Section 5, we show that the weighted cluster deletion problem is NP-complete on interval graphs and, moreover, on complete split graphs. On the positive side, we show that the unweighted problem is polynomial-time solvable for proper interval graphs, and that the weighted cluster deletion problem is polynomial-time solvable on a subclass of proper interval graphs known as paths of cliques. As a corollary of the complexity on complete split graphs, it is shown that the weighted cluster deletion problem is NP-complete on cographs, in contrast to the polynomial-time algorithm given in [9] for the unweighted case.

We conclude this introduction with some definitions. Let $G = (V, E)$ be a graph. For each

vertex $v \in V$, we denote by $N(v) = \{u : vu \in E\}$ the set of neighbors of v in G. Two vertices v and w are called *true twins* if $N(v) \cup \{v\} = N(w) \cup \{w\}$. A graph G is called *weighted* if there is a weight function $w : E \to \mathbb{R}^+$ associated with it. For the algorithms involving weighted graphs, we will assume that the weights are rational (or belong to any ordered field in which we can perform the operations algorithmically). An *unweighted* graph is a graph in which each edge has a weight equal to 1. A graph is *chordal*, if and only if it does not contain a cycle of length at least four as an induced subgraph. Given a vertex partition $S = C_1, \ldots, C_k$ of a graph G, we call the *weight* of S to the sum of the weight of all edges $e = uv$ such that $u \in C_i$, $v \in C_j$, with $i \neq j$. The cluster deletion problem of a (un)weighted graph G can be redefined as the problem of finding a clique partition of G with minimum weight.

2 Split graphs

A graph $G = (V, E)$ is a *split graph*, if and only if there is a partition of the vertices V of G into a clique K and an independent set I. Another necessary and sufficient condition for a graph G to be a split graph is that G and its complement $\overline{G}$ are chordal graphs (see [8]).

In this section we prove the NP-completeness of the weighted cluster deletion problem for split graphs by reduction from the *exact cover by 3-sets* problem (*X3C* problem for short). The formal definition of the X3C problem can be stated as follows:

Instance: A set X with $3q$ elements and a collection C of 3-element subsets of X.

Question: Does C contain an exact cover for X? In other words, is there a subset $C' \subseteq C$, such that every element of X occurs in exactly one member of C'?

The X3C problem is known to be NP-complete [10].

We analyze in this section a subclass of the split graphs, namely the class of those split graphs where each vertex of the independent set I is adjacent to exactly p vertices of the clique K. We call these graphs the p-split graphs, where $p \geq 1$.

Theorem 1 (NP-completeness on weighted 3-split graphs)**.** *The cluster deletion problem is NP-complete for weighted* 3*-split graphs, even if the weight of all the internal edges of the clique is* 1 *and the weight of the edges between the clique and the stable set is uniform.*

Proof. It is not hard to see that the cluster deletion problem is in NP, since we can easily verify in polynomial time whether a vertex partition of a graph is a clique partition or not, and if it has weight less that a given value k. Let $X = \{x_1, \ldots, x_{3q}\}$ and $C = \{c_1, \ldots, c_m\}$ be an instance of the X3C problem, where each element $c_i \in C$ is a 3-element subset of X, with $m \geq q \geq 1$. We want to know if there exists a subset $C' \subseteq C$ with size q such that each element in X belongs to exactly one of the elements in C'. We will construct an edge-weighted split graph $G = (K_X \cup I_C, E)$, where K_X induces a clique with $3q$ vertices and I_C induces an independent set with m vertices. In fact, for each element $x_i \in X$ there is a vertex v_{x_i} in K_X and for each 3-subset $c_j \in C$ there is a vertex $v_{c_j} \in I_C$. The edge set E is formed as follows: $E = E_X \cup E_C$, where $E_X = \{v_{x_i}v_{x_j} : i \neq j, v_{x_i}, v_{x_j} \in K_X\}$ and $E_C = \{v_{c_j}v_{x_{j_1}}, v_{c_j}v_{x_{j_2}}, v_{c_j}v_{x_{j_3}} : v_{c_j} \in I_C, v_{x_{j_1}}, v_{x_{j_2}}, v_{x_{j_3}} \in K_X, c_j \in C \text{ and } c_j = \{x_{j_1}, x_{j_2}, x_{j_3}\}\}$. Finally, the weight of each edge in E_X is equal to 1 and the weight of each edge in E_C is equal to

β, where $\beta = 3q$. Clearly, G is an split graph and its construction is done in polynomial time. Let $W = \binom{3q}{2} - 3q + 3(m-q)\beta$. We will show that there exists a subset $C' \subseteq C$, with $|C'| = q$, exactly covering X if and only if G has a clique partition where the sum of the weights of edges outside the cliques is at most W. In other words, there exists a solution for the X3C problem if and only if there exists an optimal solution for the cluster deletion problem for G with weight at most W.

Assume first that there exists $C' \subseteq C$, with $C' = \{c'_1, \ldots, c'_q\}$ such that $c'_i \cap c'_j = \emptyset$ whenever $i \neq j$, and $\bigcup_{c'_j \in C'} c'_j = X$. The clique partition for G can be constructed as follows: for each $c'_j \in C'$, with $c'_j = \{x_{j_1}, x_{j_2}, x_{j_3}\}$, choose the clique $\{v_{c'_j}, v_{x_{j_1}}, v_{x_{j_2}}, v_{x_{j_3}}\}$ in G. Each one of the remaining $m - q$ vertices in I_C form a clique with size one. It is easy to see that the sum of the edge weights outside those cliques is exactly equal to W.

Conversely, assume that G has a clique partition with weight at most W. We should to prove that there exists $C' \subseteq C$, with $|C'| = q$, such that C' is an exact cover for X. For this, we analyze first the structure of the optimal solutions of the cluster deletion problem for G.

Claim 1. *Let S be a clique partition of G which is an optimal solution for the cluster deletion problem in G. If there exists a clique in S formed only by vertices in K_X then, there is no solution to the X3C problem.*

First, notice that it is not possible to have two cliques, say $A_1, A_2 \in S$, formed only by vertices in K_X. Otherwise, set $S' = S \setminus \{A_1, A_2\} \cup \{A_1 \cup A_2\}$. Clearly, the weight of S' is $|A_1|.|A_2| \geq 1$ less than the weight of S, which is a contradiction to the optimally of S. On the other hand, every clique of S contains at most one vertex of I, so there are m cliques (possibly singletons) containing one vertex of I each. Let $S = A_1, B_1, \ldots, B_m$ be the clique partition of G and let W_S be the weight of S. Assume that the clique A_1 is formed only by vertices in K_X. Clearly, each clique B_j, $1 \leq j \leq m$, contains a vertex in I_C and zero, one, two or three vertices in K_X. We consider the following cases:

- Suppose there exists $B_j = \{v_{c_j}, v_{x_{j_1}}, v_{x_{j_2}}\}$ in S such that there is $v_y \in A_1$ with v_y be adjacent to v_{c_j}, where $v_{c_j} \in I_C$ and $v_y, v_{x_{j_1}}, v_{x_{j_2}} \in K_X$. Let S' be the clique partition obtained by setting $B'_j \leftarrow B_j \cup \{v_y\}$ and $A'_1 \leftarrow A_1 \setminus \{v_y\}$, where $S' = S \setminus \{A_1, B_j\} \cup \{A'_1, B'_j\}$. Let $W_{S'}$ be the weight of S'. It is not difficult to see that $W_{S'} = W_S - (\beta + 2) + (|A_1| - 1) = W_S + |A_1| - \beta - 3$. However, $\beta = 3q$ and $|A_1| \leq 3q - 2$. Therefore, $W_{S'} < W_S$, which contradicts the optimality of S.

- Similarly to the previous case, suppose there exists $B_j = \{v_{c_j}, v_{x_{j_1}}\}$ (resp. $B_j = \{v_{c_j}\}$) in S such that there is $v_y \in A_1$ with v_y be adjacent to v_{c_j}, where $v_{c_j} \in I_C$ and $v_y, v_{x_{j_1}} \in K_X$ (resp. $v_y \in K_X$) . By setting $B'_j \leftarrow B_j \cup \{v_y\}$ and $A'_1 \leftarrow A_1 \setminus \{v_y\}$, we obtain a new clique partition $S' = S \setminus \{A_1, B_j\} \cup \{A'_1, B'_j\}$ which weight $W_{S'} = W_S + |A_1| - \beta - 2 < W_S$ (resp. $W_{S'} = W_S + |A_1| - \beta - 1 < W_S$), which is a contradiction to the optimality of S.

As S is an optimal solution, no vertex in I_C is adjacent to any vertex in A_1, but by construction, it implies that X is not covered by C, and thus, there is no solution for the X3C problem. This ends the proof of this claim. $\diamondsuit$

By the previous claim, if $\bigcup_{c_j \in C} c_j = X$ then, an optimal solution S for the cluster deletion problem of G must be of the form $B_1, \ldots, B_m$, where each clique B_j contains exactly one vertex $v_{c_j} \in I_C$, for $1 \leq j \leq m$. Now, we can deduce the following result on S.

Claim 2. *Let $S = B_1, \ldots, B_m$ be an optimal solution of the cluster deletion problem for G. Let $m = t_1 + t_2 + t_3 + t_4$, with $t_i \geq 0$, for $1 \leq i \leq 4$, where t_1 (resp. t_2, t_3, t_4) represents the number of cliques B_j in S formed by a vertex in I_C and by three (resp. two, one, zero) vertices in K_X. Then, (i) $t_1 > 0$; (ii) Let v_{c_j} be a vertex in I_C such that it belongs to a clique B_j of S with $|B_j| = k$, with $1 < k \leq 3$. Then, for each vertex $x \in N(v_{c_j})$ such that $x \notin B_j$, we have that x must belong to another clique B_i of S with $|B_i| > k$.*

Suppose that $t_1 = 0$. If $t_2 \neq 0$ then, there is a clique $B_j = \{v_{c_j}, v_{x_{j_1}}, v_{x_{j_2}}\}$ in S. By construction, v_{c_j} has another neighbor $v_y \in K_X$. If $v_y \in B_i$, with $|B_i| = 3$ (resp. $|B_i| = 2$) then, we can construct another clique partition S' by setting $B'_j \leftarrow B_j \cup \{v_y\}$ and $B'_i \leftarrow B_i \setminus \{v_y\}$ and where $S' = (S \setminus \{B_i, B_j\}) \cup \{B'_j \cup B'_i\}$. Notice that the weight $W_{S'}$ of S' is equal to $W_S - (\beta+2) + \beta + 1 = W_S - 1 < W_S$ (resp. $W_S - (\beta + 2) + \beta = W_S - 2 < W_S$) which contradicts the optimality of S. If $t_1 = t_2 = 0$ then, there are $B_j, B_i, B_s \in S$ such that $B_j = \{v_{c_j}, v_{x_j}\}$, $B_i = \{v_{c_i}, v_{x_i}\}$ and $B_s = \{v_{c_s}, v_{x_s}\}$ where $\{v_{x_j}, v_{x_i}, v_{x_s}\}$ are the three neighbors of v_{c_j}. By setting $B'_j \leftarrow B_j \cup \{v_{x_i}, v_{x_j}\}$, $B'_i \leftarrow B_i \setminus \{v_{x_i}\}$, $B'_s \leftarrow B_s \setminus \{v_{x_s}\}$, and by constructing the partition $S' = S \setminus \{B_j, B_i, B_s\} \cup \{B'_j, B'_i, B'_s\}$ we have that the weight $W_{S'}$ of S' is equal to $W_S - (2\beta + 3) + 2\beta = W_S - 3 < W_S$, which is again a contradiction to the optimality of S. Therefore, $t_1 > 0$ and thus (i) holds. In an analogous way, we can show (ii), which ends the proof of this claim. $\diamondsuit$

Now, let $S = B_1, \ldots, B_m$ be an optimal clique partition of G with weight W_S. By Claim 1 we know that each clique B_i in S contains exactly one vertex in I_C and that there are t_1 (res. t_2, t_3, t_4) cliques in S having exactly three (resp. two, one, zero) vertices in K_X. By Claim 2, we know that $t_1 > 0$ and $t_2, t_3, t_4 \geq 0$. Therefore, the weight W_S of S can be written as: $W_S = \binom{3q}{2} - (3t_1 + t_2) + (t_2 + 2t_3 + 3t_4)\beta$. Moreover, $m = t_1 + t_2 + t_3 + t_4$ and $3q = 3t_1 + 2t_2 + t_3$. It is not difficult to see that $3(m - q) = t_2 + 2t_3 + 3t_4$ and that $3t_1 + t_2 = 3q - t_2 - t3$ which implies that $3q \geq 3t_1 + t_2$. Therefore, $W \leq W_S$ with equality only if $t_2 = t_3 = 0$ in which case, $t_1 = q$. Therefore, if S has weight W then, there is a solution to the X3C problem which ends the proof of this theorem. $\square$

For the unweighted case, we will show next that the problem can be easily solved on split graphs.

Theorem 2 (Polynomiality on unweighted split graphs)**.** *The cluster deletion problem is polynomial-time solvable for unweighted split graphs. Indeed, if (K, I) is a split partition of the graph G such that K is a maximal clique of G, then $\{K\} \cup \{\{v\} : v \in I\}$ is an optimal solution unless there is a vertex v_1 in I adjacent to all but one vertex w in K, and that vertex w has a neighbor v_2 in I. In that case, an optimal solution is $\{\{v_1\} \cup (K - \{w\}), \{w, v_2\}\} \cup \{\{v\} : v \in I, v \neq v_1, v_2\}$.*

Proof. Let G be a split graph and let (K, I) be a split partition of G such that K is a maximal clique of G. We can see the problem as the problem of maximizing the internal edges in a clique partition S (it is equivalent to the problem of minimizing the edges joining distinct cliques of S). In order to break ties, we will maximize the internal edges that join two vertices of K in G. We will call the *standard partition* to $S_0 = \{K\} \cup \{\{v\} : v \in I\}$.

Suppose that we have an optimal solution S with respect to the criterion that is strictly better than S_0. If S contains a clique $A \subsetneq K$ such that $|A| = a \geq 1$ and a clique B containing one vertex of I and $b \geq 1$ vertices of K, then $S' = S \setminus \{A, B\} \cup \{A \cup (B \cap K), B \cap I\}$, with respect to S, has ab new internal edges that join two vertices of K in G and looses b edges that join vertices of K with vertices of I, so it is either strictly better than S or it is preferable by the breaking ties criterion, a

contradiction to the optimality of S. So, we may assume that every clique of S contains a vertex of I.

Suppose now that the clique A has a vertex of I and $a \geq 2$ vertices of K, and the clique B has a vertex of I and $b \geq 2$ vertices of K. Then $S' = S \setminus \{A, B\} \cup \{(A \cap K) \cup (B \cap K), A \cap I, B \cap I\}$, with respect to S, has ab new internal edges that join two vertices of K in G and looses $a+b$ edges that join vertices of K with vertices of I, so as a and b are at least 2, S' is either strictly better than S or it is preferable by the breaking ties criterion, a contradiction to the optimality of S.

Suppose now that there are three cliques A, B, C having a vertex of I each and $a, b, c \geq 1$ vertices of K, respectively. Then $S' = S \setminus \{A, B, C\} \cup \{(A \cap K) \cup (B \cap K) \cup (C \cap K), A \cap I, B \cap I, C \cap I\}$, with respect to S, has $ab + ac + bc$ new internal edges that join two vertices of K in G and looses $a + b + c$ edges that join vertices of K with vertices of I, so as $ab \geq a$, $bc \geq b$, and $ca \geq c$, S' is either strictly better than S or it is preferable by the breaking ties criterion, a contradiction to the optimality of S.

Then S has exactly two cliques containing vertices of K, each of them contains a vertex of I, and at least one of them has only one vertex of K. So, there is a vertex v_1 in I adjacent to all but one vertex w in K, and that vertex w has a neighbor v_2 in I, and $S = \{\{v_1\} \cup (K - \{w\}), \{w, v_2\}\} \cup \{\{v\} : v \in I, v \neq v_1, v_2\}$. □

3 Chordal graphs

Chordal graphs form an extensively studied class of graphs, as their peculiar clique-based structure allows an efficient solution for algorithmic problems. A very nice survey by Blair and Peyton on chordal graphs properties can be found in [2].

In order to prove the main result of this section, we will show first a simple general lemma.

Lemma 1 (True twins)**.** *Let G be graph and v, w be true twins in G. Then, for every optimal solution of the unweighted cluster deletion problem, v and w belong to the same clique of the partition.*

Proof. Suppose, on the contrary, that there is an optimal clique partition S such that v belongs to a clique C_1 and w to a different clique C_2. Without loss of generality, we may assume $|C_1| \leq |C_2|$. But then $S' = S \setminus \{C_1, C_2\} \cup \{C_1 \setminus \{v\}, C_2 \cup \{v\}\}$ has strictly less weight than S, a contradiction. □

Theorem 3 (NP-completeness on unweighted chordal graphs)**.** *The cluster deletion problem is NP-complete for unweighted P_5-free chordal graphs.*

Proof. The proof of this theorem is similar to the proof of Theorem 1. The reduction is done again from the X3C problem. Let $X = \{x_1, \ldots, x_{3q}\}$ and $C = \{c_1, \ldots, c_m\}$ be an instance of the X3C problem, where each element $c_i \in C$ is a 3-element subset of X, with $m \geq q \geq 1$. We want to know if there exists a subset $C' \subseteq C$ with size q such that each element in X belongs to exactly one of the elements in C'. We construct a graph $G = (V, E)$ as follows. The set of vertices V is formed by the subsets $K_X, K_{c_1}, \ldots, K_{c_m}$, where $K_X = \{v_{x_i} : x_i \in X\}$ is the subset of vertices representing the elements of X. Let $\beta = 3q$. For each 3-subset $c_i \in C$, we construct a set of vertices $K_{c_i} = \{v_{c_{i,1}}, \ldots, v_{c_{i,\beta}}\}$ with cardinality β. The set of edges is constructed as follows: K_X induces a clique of size β, and also each K_{c_i} induces a clique of size β, for $1 \leq i \leq m$. Finally, for each $c_i \in C$ such that $c_i = \{x_{i_1}, x_{i_2}, x_{i_3}\}$, we add all the edges between the vertices of the clique K_{c_i} to the vertices $v_{x_{i_1}}, v_{x_{i_2}}, v_{x_{i_3}} \in K_X$. Clearly, this construction is done in polynomial time and the

obtained graph G is an unweighted P_5-free chordal graph. Now, since for each $1 \leq i \leq m$, all the β vertices of the clique K_{c_i} are true twins, by Lemma 1 they must belong to the same clique in an optimal partition S. Therefore, we can contract each clique K_{c_i}, with $1 \leq i \leq m$, into a single vertex c_i and replace each subset of β edges between K_{c_i} and the vertex $v_{x_{i_j}} \in K_X$ by a single edge with weight β, where $1 \leq j \leq 3$ and $x_{i_j} \in c_i$. This is exactly the weighted split graph used in Theorem 1. □

4 Block graphs

A graph G is a *block* graph if it is a connected graph and every *block* (i.e. maximal 2-connected component) is a clique. Block graphs are a subclass of chordal graphs.

The first result in this section concerns weighted 1-split graphs, a particular subclass of block graphs. We will use submodular functions minimization in order to solve the cluster deletion problem on 1-split graphs, and then we will reduce the problem on weighted block graphs to the problem on weighted 1-split graphs.

Given a finite nonempty set V of cardinality n, a function f defined on all the subsets of V is called *submodular* if it satisfies $f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$, for every $X, Y \subseteq V$. In [13] and [16] are presented combinatorial polynomial-time algorithms for finding a minimizer of a general submodular function, provided that an oracle for evaluating the function value is available. The number of oracle calls is bounded by a polynomial in the size of the underlying set.

Theorem 4 (Polynomiality on weighted 1-split graphs)**.** *Cluster deletion is polynomial time solvable for weighted* 1*-split graphs, provided that the weights of the internal edges of the clique are nonnegative.*

Proof. In the cluster deletion problem of a 1-split graph whose split partition is (K, I), the solution has at most one clique K_1 that has no intersection with I, some pairs $\{v, u\}$ with $v \in K$ and $u \in I$, and some isolated vertices from I. Each vertex of I has at most one neighbor in K, and if $\{v, u\}$ belongs to the solution, then u is one of the neighbors of v in I maximizing the weight of the edge vu. Indeed, we can preprocess the graph, identifying a subset of vertices of I that will be trivial cliques in the solution, and leaving a graph in which each vertex of K has at most one neighbor in I. If the subset K_0 of vertices of K having no neighbors in I is nonempty, it will be part of the clique K_1 in the solution. We will name the set $K \setminus K_0$ as $\{v_1, \ldots, v_r\}$ and their respective neighbors in I as $\{u_1, \ldots, u_r\}$. So, a candidate solution is totally determined by the subset S of $\{1, \ldots, r\}$ such that the solution is $\{\{v_i, u_i\}\}_{i \notin S} \cup \{K_1 \cup \{v_i : i \in S\}\} \cup \{\{u_i\}\}_{i \in S}$. The subset S is not necessarily a proper subset of $\{1, \ldots, r\}$ and it could be empty. We will prove that the function f that assigns to S the difference of weight between the candidate solution associated with S and the solution associated with the empty set is submodular, and thus the set S that minimizes that function can be found in polynomial time. The function f can be computed in polynomial time for a subset S, and is defined as

$$f(S) = \sum_{i \in S} w(v_i u_i) - \sum_{i,j \in S} w(v_i v_j) - \sum_{i \in S, z \in K_0} w(v_i z)$$

Let $S, T \subseteq \{1, \ldots, r\}$. We will show that $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$, provided that the weights of the edges joining two vertices of K are nonnegative. It is easy to see that $\sum_{i \in S} w(v_i u_i) + \sum_{i \in T} w(v_i u_i) = \sum_{i \in S \cup T} w(v_i u_i) + \sum_{i \in S \cap T} w(v_i u_i)$ and that $\sum_{i \in S, z \in K_0} w(v_i z) +$

$\sum_{i\in T,z\in K_0} w(v_iz) = \sum_{i\in S\cup T,z\in K_0} w(v_iz) + \sum_{i\in S\cap T,z\in K_0} w(v_iz)$. So we have to show that $\sum_{i,j\in S} w(v_iv_j) + \sum_{i,j\in T} w(v_iv_j) \leq \sum_{i,j\in S\cup T} w(v_iv_j) + \sum_{i,j\in S\cap T} w(v_iv_j)$, but $\sum_{i,j\in S\cup T} w(v_iv_j) + \sum_{i,j\in S\cap T} w(v_iv_j) - \sum_{i,j\in S} w(v_iv_j) - \sum_{i,j\in T} w(v_iv_j) = \sum_{i\in S\setminus T,j\in T\setminus S} w(v_iv_j) \geq 0$.

If the weight of the internal edges is 1, the algorithm is very simple. We name the vertices in $K \setminus K_0$ as $\{v_1, \ldots, v_r\}$ and their respective neighbors in I as $\{u_1, \ldots, u_r\}$, in such a way that $w(v_1u_1) \leq \cdots \leq w(v_ru_r)$. Then the only sets that are candidates to minimize f are the empty set and the sets $\{1, \ldots, j\}$, for $1 \leq j \leq r$. □

Based on the solution of cluster deletion for weighted 1-split graphs, we will solve the problem for weighted block graphs, when the weight of the edges is nonnegative.

Theorem 5 (Polynomiality on weighted block graphs)**.** *Cluster deletion is polynomial time solvable for weighted block graphs, when the weight of the edges is nonnegative.*

Proof. Let $G = (V, E)$ be a weighted block graph, such that the weight w of the edges is nonnegative. An *end-block* of a graph is a block containing exactly one cut-vertex. It is known that every connected graph that is not 2-connected has an end-block. Inductively, the blocks of G can be enumerated as $B_1, \ldots, B_r$ in such a way that B_i is an end-block of the graph induced by $B_i \cup B_{i+1} \cup \cdots \cup B_r$. We will process the blocks in that order by solving each time a subproblem and reducing iteratively the graph to a simpler graph. Then we will reconstruct the solution for the original graph based on the solution of each subproblem and the recursive solution of the reduced graph.

By the way of choosing the order of the blocks, when processing the block B_i for $i < r$, it will have exactly one cut-vertex v joining B_i with non-processed blocks. The graph G_0 will be G, we will create a graph G_i from G_{i-1} by replacing the connected component of $G_{i-1} \setminus \{v\}$ that contains $B_i \setminus \{v\}$ by a vertex u adjacent to v and we will assign to the edge vu a suitable weight. It means, inductively, that when processing block B_i, the connected component H of $G_{i-1} \setminus \{v\}$ that contains $B_i \setminus \{v\}$ is a 1-split graph, and so is the graph induced by $V(H) \cup \{v\}$. We will define $w(uv) = \sum_{y\in H,vy\in E} w(vy) + w(S_H) - w(S_{H\cup\{v\}})$, where $S_{H\cup\{v\}}$ and S_H are optimal solutions for the cluster deletion problem on the 1-split graph induced by $V(H) \cup \{v\}$, and H, respectively. Notice that we can find $S_{H\cup\{v\}}$ and S_H by Theorem 4.

Claim 3. *Let S_i be an optimal solution of the cluster deletion problem for G_i. If $\{u, v\} \in S_i$, then $S_i \setminus \{\{u, v\}\} \cup S_{H\cup\{v\}}$ is an optimal solution for G_{i-1}. If $\{u, v\} \notin S_i$, then $S_i \setminus \{\{u\}\} \cup S_H$ is an optimal solution for G_{i-1}.*

Let S_{i-1} be an optimal solution of the cluster deletion problem for G_{i-1}. Let S^1_{i-1} the subset of cliques of S_{i-1} containing vertices of H and $S^2_{i-1} = S_{i-1} \setminus S^1_{i-1}$. We have that either v belongs to a clique in S^1_{i-1} or not. As v is a cut-vertex of G_{i-1}, it is clear that in the first case S^1_{i-1} is an optimal solution for the cluster deletion problem on the graph induced by $V(H) \cup \{v\}$, while in the second case S^1_{i-1} is an optimal solution for the cluster deletion problem on the graph H. On the other hand, in the first case, $S^2_{i-1} \cup \{v, u\}$ will be a feasible solution for G_i, with weight $w(S_{i-1}) - w(S_{H\cup\{v\}})$, while in the second case, $S^2_{i-1} \cup \{u\}$ will be a feasible solution for G_i, with weight $w(S_{i-1}) - w(S_H) - \sum_{y\in H,vy\in E} w(vy) + w(uv) = w(S_{i-1}) - w(S_{H\cup\{v\}})$. So, if S_i is an optimal solution of the cluster deletion problem for G_i, then $w(S_i) \leq w(S_{i-1}) - w(S_{H\cup\{v\}})$, which implies $w(S_i) + w(S_{H\cup\{v\}}) \leq w(S_{i-1})$.

Now, let us see that the weight of the solutions proposed for G_{i-1} in each case is exactly $w(S_i) + w(S_{H\cup\{v\}})$, which makes them respectively optimal.

If $\{u,v\} \in S_i$, it is easy to see that $w(S_i \setminus \{\{u,v\}\} \cup S_{H\cup\{v\}}) = w(S_i) + w(S_{H\cup\{v\}})$. If $\{u,v\} \notin S_i$, when considering $S_i \setminus \{\{u\}\} \cup S_H$ in G_{i-1}, we do not have to delete the edge uv, but we have to delete every edge joining v with vertices in H, so $w(S_i \setminus \{\{u\}\} \cup S_H) = w(S_i) - w(uv) + \sum_{y\in H, vy\in E} w(vy) + w(S_H) = w(S_i) + w(S_{H\cup\{v\}})$. This ends the proof of this claim. $\diamondsuit$

Finally, when processing block B_r, the graph G_{r-1} is completely reduced to a 1-split graph, so we can also apply the algorithm of Theorem 4 in order to obtain an optimal partition for G_{r-1}, and then recursively construct the solution for the graph $G_0 = G$, following the ideas in the Claim.

If the graph G is unweighted, then the 1-split graphs in which we need to solve the subproblems have weight 1 in every internal edge. In that case, as we noticed in Theorem 4, the algorithm is very simple. $\square$

5 Interval graphs

Other interesting subclasses of chordal graphs are interval graphs. A graph G is an *interval graph* if G is the intersection graph of a finite family of intervals of the real line, and it is a *proper interval graph* if it admits an intersection model in which no interval properly contains another. A *unit interval graph* is the intersection graph of a finite family of intervals of the real line, all of the same length. Proper interval graphs and unit interval graphs coincide, and they are exactly the claw-free interval graphs [15] (the *claw* is the complete bipartite graph $K_{1,3}$).

A restricted subclass of unit interval graphs is the class of *paths of cliques*. A graph is a *path of cliques* if after identifying true twins into a single vertex, the resulting graph is a single path. That is, its vertex set can be partitioned into sets $A_1, \ldots, A_n$ in such a way that for each pair of vertices v, w, such that $v \in A_i$ and $w \in A_j$, they are adjacent if and only if either $i = j$ or $i = j + 1$ or $i = j - 1$. They also appear in the literature as *line graphs of multipaths*.

We will prove the next result with an approach similar to the proof od Theorem 4. We will define an initial solution and then we will code every other solution by a subset of a set. Finally, we will show that the function that assigns to every subset the difference of weights between its associated solution and the initial one is submodular.

Theorem 6 (Polynomiality on weighted paths of cliques)**.** *The cluster deletion problem is polynomial-time solvable for weighted paths of cliques, when the weights are nonnegative.*

Proof. Let $A_1, \ldots, A_n$ be the vertex set partition of a path of cliques G. Without loss of generality we can assume that n is even, by adding if necessary a set A_{n+1} with only one vertex adjacent to every vertex in A_n with edges of weight zero. Let $n = 2r$ and define the initial solution as the cliques $A_{2k-1} \cup A_{2k}$ for $1 \leq k \leq r$. Every vertex of A_i, for $1 < i < n$, has two possibilities. Either it will be part of a clique contained in $A_{i-1} \cup A_i$ or it will be part of a clique contained in $A_i \cup A_{i+1}$. So, an arbitrary solution is completely defined by the subset S of $V(G) \setminus (A_1 \cup A_n)$ that, from the initial solution, are moved into a clique contained in $A_{2j} \cup A_{2j+1}$ for some integer $1 \leq j \leq r - 1$. We will define $f(S)$ for a subset S of $V(G) \setminus (A_1 \cup A_n)$ as the difference between the weight of the solution associated with S and the weight of the initial solution. The optimal solution will be given by the subset S that minimizes function f. We will show now that S is a submodular function and so a minimizer can be found in strongly polynomial time.

The function f can be expressed as $f(S) = \sum_{1 \leq k \leq r} f^k(S)$, where

$$f^k(S) = \sum_{\substack{v \in S \cap A_{2k} \\ u \in A_{2k-1}}} w(vu) + \sum_{\substack{v \in S \cap A_{2k-1} \\ u \in A_{2k}}} w(vu) + \sum_{\substack{v \in S \cap A_{2k-1} \\ u \in A_{2k-1} \setminus S}} w(vu) + \sum_{\substack{v \in S \cap A_{2k} \\ u \in A_{2k} \setminus S}} w(vu) - \sum_{\substack{v \in S \cap A_{2k} \\ u \in S \cap A_{2k+1}}} w(vu) - \sum_{\substack{v \in S \cap A_{2k-1} \\ u \in S \cap A_{2k}}} w(vu)$$

where in order to simplify the expression, we let $A_{2r+1} = \emptyset$. Notice that the last term in the sum avoids the double counting of the first two terms.

For each value of k, define $f_1^k(S), \ldots, f_6^k(S)$ as the six terms of $f^k(S)$. It is easy to see that for $i = 1, 2$ and S, T subsets of $V(G) \setminus (A_1 \cup A_n)$, it holds $f_i^k(S) + f_i^k(T) = f_i^k(S \cup T) + f_i^k(S \cap T)$. We will prove that, for $i = 3, \ldots, 6$, $f_i^k(S) + f_i^k(T) \geq f_i^k(S \cup T) + f_i^k(S \cap T)$.

For S, T subsets of $V(G) \setminus (A_1 \cup A_n)$, and by decomposing S as $(S \setminus T) \cup (S \cap T)$ (resp. T as $(T \setminus S) \cup (S \cap T)$); $V(G) \setminus S$ as $(T \setminus S) \cup (V(G) \setminus (S \cup T))$ (resp. $V(G) \setminus T$ as $(S \setminus T) \cup (V(G) \setminus (S \cup T))$); $S \cup T$ as $(S \setminus T) \cup (T \setminus S) \cup (S \cap T)$; and $V(G) \setminus (S \cap T)$ as $(T \setminus S) \cup (S \setminus T) \cup (V(G) \setminus (S \cup T))$, it can be see that

$$f_3^k(S) + f_3^k(T) - f_3^k(S \cup T) - f_3^k(S \cap T) = 2 \sum_{\substack{v \in (S \setminus T) \cap A_{2k-1} \\ u \in (T \setminus S) \cap A_{2k-1}}} w(vu) \geq 0$$

because the weights are nonnegative, so f_3^k is submodular. The proof for f_4^k is identical.

Recall that $f_5^k(S) = -\sum_{v \in S \cap A_{2k}, u \in S \cap A_{2k+1}} w(vu)$. By using again the decomposition S as $(S \setminus T) \cup (S \cap T)$ (resp. T as $(T \setminus S) \cup (S \cap T)$); and $S \cup T$ as $(S \setminus T) \cup (T \setminus S) \cup (S \cap T)$, it can be see that

$$f_5^k(S) + f_5^k(T) - f_5^k(S \cup T) - f_5^k(S \cap T) = \sum_{\substack{v \in (S \setminus T) \cap A_{2k} \\ u \in (T \setminus S) \cap A_{2k+1}}} w(vu) + \sum_{\substack{v \in (T \setminus S) \cap A_{2k} \\ u \in (S \setminus T) \cap A_{2k+1}}} w(vu) \geq 0$$

because the weights are nonnegative, so f_5^k is submodular. The proof for f_6^k is identical. Finally, the sum of submodular functions is submodular, and this completes the proof. □

A split graph is called *complete* if each vertex of the stable set is adjacent to all vertices of the clique. By slightly modifying the proof of Theorem 1, we can prove the following.

Theorem 7 (NP-completeness on weighted complete split graphs)**.** *The cluster deletion problem is NP-complete for weighted complete split graphs, even if the weight of all the internal edges of the clique is* 1*.*

Proof. We will reduce once more time the X3C problem. Let $X = \{x_1, \ldots, x_{3q}\}$ and $C = \{c_1, \ldots, c_m\}$ be an instance of the X3C problem, where each element $c_i \in C$ is a 3-element subset of X, with $m \geq q \geq 1$. We want to know if there exists a subset $C' \subseteq C$ with size q such that each element in X belongs to exactly one of the elements in C'. We will construct an edge-weighted complete split graph $G = (K_X \cup I_C, E)$, where K_X induces a clique with $3q$ vertices and I_C induces an independent set with m vertices. In fact, for each element $x_i \in X$ there is a vertex v_{x_i} in K_X and for each 3-subset $c_j \in C$ there is a vertex $v_{c_j} \in I_C$. The edge set E is formed as follows: $E = E_X \cup E_C \cup E_{\bar{C}}$, where $E_X = \{v_{x_i} v_{x_j} : i \neq j, v_{x_i}, v_{x_j} \in K_X\}$, $E_C = \{v_{c_j} v_{x_{j_1}}, v_{c_j} v_{x_{j_2}}, v_{c_j} v_{x_{j_3}} : v_{c_j} \in I_C, v_{x_{j_1}}, v_{x_{j_2}}, v_{x_{j_3}} \in K_X, c_j \in C \text{ and } c_j = \{x_{j_1}, x_{j_2}, x_{j_3}\}\}$,

and $E_{\bar{C}} = \{v_{c_j}v_{x_i} : v_{c_j} \in I_C, v_{x_i} \in K_X, i \notin \{j_1, j_2, j_3\}, c_j \in C \text{ and } c_j = \{x_{j_1}, x_{j_2}, x_{j_3}\}\}$. Finally, the weight of each edge in $E_X \cup E_{\bar{C}}$ is equal to 1, and the weight of each edge in E_C is equal to $\beta = \binom{3q}{2} + 3m(q-1) + 1$. Clearly, G is a complete split graph and its construction is done in polynomial time. Let $W = \binom{3q}{2} - 3q + 3(m-q)\beta + 3m(q-1)$. We will show that there exists a subset $C' \subseteq C$, with $|C'| = q$, exactly covering X if and only if G has a clique partition where the sum of the weights of edges outside the cliques is at most W. In other words, there exists a solution for the X3C problem if and only if there exists an optimal solution for the cluster deletion problem for G with weight at most W.

Assume first that there exists $C' \subseteq C$, with $C' = \{c'_1, \ldots, c'_q\}$ such that $c'_i \cap c'_j = \emptyset$ whenever $i \neq j$, and $\bigcup_{c'_j \in C'} c'_j = X$. The clique partition for G can be constructed as follows: for each $c'_j \in C'$, with $c'_j = \{x_{j_1}, x_{j_2}, x_{j_3}\}$, choose the clique $\{v_{c'_j}, v_{x_{j_1}}, v_{x_{j_2}}, v_{x_{j_3}}\}$ in G. Each one of the remaining $m - q$ vertices in I_C form a clique with size one. It is easy to see that the sum of the edge weights outside those cliques is exactly equal to W.

Conversely, assume that G has a clique partition with weight at most W. We should to prove that there exists $C' \subseteq C$, with $|C'| = q$, such that C' is an exact cover for X. For this, we analyze first the structure of such a solution. Notice that β is greater than the sum of the weights of all the edges in $E_X \cup E_{\bar{C}}$, so a solution with weight W should delete exactly $3(m-q)$ edges of E_C. It cannot delete less than $3(m-q)$ because each vertex of K_X can be in the same clique with at most one vertex of I_C. So, every vertex of K_X is in a clique with one vertex of I_C, and joined to that vertex by an edge of the set E_C. Than means that all the edges of $E_{\bar{C}}$ are deleted and that each clique contains at most three vertices of K_X. By the value of W, we can see that each clique should contain exactly three vertices of K_X, and following the lines of the proof of Theorem 1, such a solution corresponds to a solution of the X3C instance, which ends the proof of this theorem. □

Complete split graphs are also interval graphs, but (in general) they are not unit interval graphs, as if the stable set is of size at least three and the complete set is nonempty, they contain a *claw* (the complete bipartite graph $K_{1,3}$), that is not a unit interval graph. So, we have the following corollary.

Corollary 1 (NP-completeness on weighted interval graphs)**.** *The cluster deletion problem is NP-complete for weighted interval graphs.*

For unweighted unit interval graphs instead, we will show that the cluster deletion problem is polynomial-time solvable. First, we need to state some known results and show a lemma describing the structure of an optimal solution.

Theorem 8 (Roberts, 1969 [15])**.** *A graph G is a unit interval graph if and only if its vertices can be linearly ordered such that, for each clique M of G, the vertices contained in M are consecutive.*

Such an ordering is called a *canonical ordering* of the vertices.

Lemma 2 (Consecutiveness for unweighted unit interval graphs)**.** *Let G be an unweighted unit interval graph and $v_1, \ldots, v_n$ be a canonical ordering of the vertices of G. Then there is an optimal solution of the cluster deletion problem for G, such that each clique of the solution consists of consecutive vertices in that ordering.*

Proof. Let us define, for each clique B of the optimal solution S, $m(B) = \max\{j : v_j \in B\}$. Now, let $B_1, \ldots, B_k$ be the cliques of the solution ordered by m increasingly. Suppose that not all the

cliques consist of consecutive vertices, and let i minimum such that either $i < m(B_1)$ and $v_i \notin B_1$, or $m(B_{j-1}) < i < m(B_j)$ but $v_i \notin B_j$, for some j. Let j' such that $v_i \in B_{j'}$. Then, by the choice of i, all the vertices of $B_{j'}$ have subindex greater than i and $j' > j$ so $m(B_{j'}) > m(B_j)$. Since v_i is adjacent to $v_{m(B_{j'})}$ and G is a proper interval graph with canonical ordering $v_1, \ldots, v_n$, the vertices $v_i, \ldots, v_{m(B_{j'})}$ form a clique and, in particular, $B_{j'} \cup \{v_{m(B_j)}\}$ is a clique and v_i is adjacent to $v_{m(B_j)}$. So, independently of i being greater os less than the minimum index of a vertex in B_j, $B_j \cup \{v_i\}$ is a clique. Then, either $|B_j| \geq |B'_j|$ and $S \setminus \{B_j, B_{j'}\} \cup \{B_j \cup \{v_i\}, B_{j'} \setminus \{v_i\}\}$ is a solution strictly better than S, or $|B_j| < |B'_j|$ and $S \setminus \{B_j, B_{j'}\} \cup \{B_j \setminus \{v_{m(B_j)}\}, B_{j'} \cup \{v_{m(B_j)}\}\}$ is a solution strictly better than S, a contradiction in both cases. □

Theorem 9 (Polynomiality on unweighted unit interval graphs)**.** *The cluster deletion problem can be solved in polynomial time on unweighted unit interval graphs.*

Proof. By using Lemma 2, we can easily develop a dynamic programming algorithm. For $i = 0, 1, \ldots, n$, let $f(i)$ be the value of an optimal cluster deletion solution for the subgraph of G induced by $v_1, \ldots, v_i$. Then $f(0) = f(1) = 0$ and, for $i > 1$, $f(i)$ will be the minimum over all the possible cliques containing v_i, i.e., $\{v_j, \ldots, v_i\}$ where either $j = i$ or $v_j v_i \in E(G)$, of $f(j-1)$ plus the amount of vertices joining $\{v_1, \ldots, v_{j-1}\}$ with $\{v_j, \ldots, v_i\}$ (this is 0 if $j = 1$). By keeping also the number j realizing the minimum $f(i)$, we can also reconstruct the partition itself. □

General interval graphs do not have the same clique structure as unit interval graphs. And for weighted unit interval graphs, Lemma 2 does not hold. An example of this is the graph P_6^2, whose vertices are $v_1, \ldots, v_6$ and v_i is adjacent to v_j if and only if $1 \leq |i - j| \leq 2$. It is easy to see that the only possible canonical orderings for P_6^2 are $v_1, \ldots, v_6$ or $v_6, v_5, \ldots, v_1$. Let w be defined on the edges of P_6^2 such that $w(v_2v_4) = w(v_3v_5) = 100$ and $w(e) = 1$ for every other edge e. Any solution to the cluster deletion problem that does not contain $\{v_2, v_4\}$ and $\{v_3, v_5\}$ as cliques has weight at least 100, so the optimal solution is to have $\{v_2, v_4\}$, $\{v_3, v_5\}$, and isolated vertices, with a weight equal to 7.

The example shows that the idea of Theorem 9 cannot be generalized in a straightforward way. It remains as an open question the computational complexity of the cluster deletion problem on unweighted interval graphs and on weighted unit interval graphs.

Complete split graphs are also cographs (i.e. P_4-free graphs, that are not necessarily chordal as they may contain C_4 as induced subgraph). The cluster deletion problem on unweighted cographs was solved in polynomial time by Gao et al. in [9]. As a corollary of Theorem 7, we have this complexity result for the weighted case.

Corollary 2 (NP-completeness on weighted cographs)**.** *The cluster deletion problem is NP-complete for weighted cographs.*

6 Further results and open problems

We showed in Theorem 1 that the cluster deletion problem is NP-complete for weighted 3-split graphs even if the weight of all the internal edges of the clique is 1 and the weight of the edges between the clique and the stable set is uniform. We have seen also in Theorem 4 that the cluster deletion problem is polynomial-time solvable for weighted 1-split graphs. As for 2-split graphs, we will show next that, under the conditions of Theorem 1 (the weight of all the internal edges of the

clique is 1 and the weight of the edges between the clique and the stable set is uniform) the problem is polynomial-time solvable.

Theorem (Polynomiality on restricted weighted 2-split graphs)**.** *The cluster deletion problem is polynomial-time solvable for weighted* 2*-split graphs, if the weight of all the internal edges of the clique is* 1 *and the weight of the edges between the clique and the stable set is uniform.*

Proof. Let G be a 2-split graph with split partition (K, I), and β be the weight of the edges between K and I. Let us create a graph G' with vertex set K', where K' is the subset of vertices of K that have at least one neighbor in I, and such that two vertices are adjacent in G' if they have a common neighbor of I in G. Minimizing the sum of the external edges of a clique partition of G is equivalent to maximizing the sum of the internal edges of the clique partition. There are three possible class of cliques in G: those containing one vertex of I and two vertices of K, those containing one vertex of I and one vertex of K, or those completely included in K, and by optimality, there is just one of such cliques. Then the sum of internal edges will be $a(2\beta + 1) + b\beta + c(c-1)/2$ $(*)$, where $0 \leq a \leq \nu(G')$ ($\nu(G')$ is the value of a maximum matching of G'), $0 \leq b \leq |K'| - 2a$, and $c = |K| - 2a - b$. It is easy to see that in an optimal solution, either $a = \nu(G')$ or $b = 0$. In the first case, after the substitution $a = \nu(G')$, the coefficient of b in the expression $(*)$ is positive, so the maximum is attained either by $b = 0$ or by $b = |K'| - 2\nu(G')$. In the second case, after the substitution $b = 0$, the coefficient of a in the expression $(*)$ is positive, so the maximum is attained either by $a = 0$ or by $a = \nu(G')$. So the problem is reduced to solve maximum matching in G' and then compute the value of $(*)$ for the three possible optimal solutions $a = 0, b = 0$; $a = \nu(G'), b = 0$; and $a = \nu(G'), b = |K'| - 2\nu(G')$. □

We leave as an open problem the computational complexity of the cluster deletion problem in general weighted 2-split graphs, or even when the weight of all the internal edges of the clique is 1 but the weight of the edges between the clique and the stable set is arbitrary and not necessarily uniform.

References

[1] N. Bansal, A. Blum, S. Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004. Extended abstract appeared in FOCS 2002, pp. 238-247.

[2] J.R.S. Blair, B. Peyton. An introduction to chordal graphs and clique trees. In: *Graph Theory and Sparse Matrix Computation*, The IMA Volumes in Mathematics and its Applications Volume 56, 1993, pp. 1–29.

[3] S. Böcker, P. Damaschke. Even faster parametrized cluster deletion and cluster editing. *Information Processing Letters*, 111:717–721, 2011.

[4] M. Charikar, V. Guruswami, A. Wirth. Clustering with qualitative information. In Proc. of *44th Annu. IEEE Symp. Foundations of Computer Science*, FOCS 2003, pp. 524-533.

[5] E. D. Demaine, D. Emanuel, A. Fiat, N. Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361:172-187, 2006.

[6] A. Dessmark, A. Lingas, E. M. Lundell, M. Persson, J. Jansson. On the approximability of maximum and minimum edge clique partitions problems. *International Journal of Foundations of Computer Science*, 18(2):217–226, 2007.

[7] J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards–B, Mathematics and Mathematical Physics*, 69B:125–130, 1965.

[8] S. Földes, P. L. Hammer. Split graphs. In *Proc. of 8th South-Eastern Conference on Combinatorics, Graph Theory and Computing*, Congressus Numerantium, 19:311–315, 1977.

[9] Y. Gao, D. R. Hare, J. Nastos. The cluster deletion problem for cographs. *Discrete Mathematics*, 313:2763–2771, 2013.

[10] M. R. Garey, D. S. Johnson. *Computers and Intractability*, Freeman, San Francisco, 1979.

[11] P. Hansen, B. Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79:191–215, 1997.

[12] J. Hartigan. *Clustering algorithms*, Wiley, New York, 1975.

[13] S. Iwata, L. Fleischer, S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48:761–777, 2001.

[14] C. Komusiewicz, J. Uhlmann. Cluster editing with locally bounded modifications. *Discrete Applied Mathematics*, 160(15):2259–2270, 2012.

[15] F.S. Roberts. Indifference graphs. In: *Proof Techniques in Graph Theory* (F. Harary, ed.), Academic Press, 1969, pp. 139–146.

[16] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80:346–355, 2000.

[17] R. Shamir, R. Sharan, D. Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1–2):173–182, 2004.